

AMENDMENTS TO THE CLAIMS

C1 Claim 1 (currently amended): A program control apparatus for controlling execution of a program in a computer system in which a plurality of threads are switched in accordance with a scheduling policy by a scheduler, comprising:

a first means ~~responsive to a predetermined first application program interface call from one of said plurality of threads for setting a prescribed flag to a first state, wherein said first means for setting a prescribed flag is responsive to a first application program interface call for requesting a start of context switching detection made from a first thread of said plurality of threads; to one of first and second states, said first means includes means responsive to an application program interface call from one of said plurality of threads which requests start of detection of presence/absence of a context switching, for setting a flag indicating presence/absence of a context switching to a state indicating absence of a context switching, and for disabling context switching between threads when said flag is set to correspond to absence of context switching;~~

a second means for setting the prescribed flag to a second state, wherein said second means for setting the prescribed flag is responsive to context switches and sets the prescribed flag to the second state when a context switch is detected while the prescribed flag is in the first state; detecting, after said flag is set to said one state, a prescribed change in a state of said scheduling policy, and for setting said flag to a other one of said first and second states, said second means includes means for setting, after said flag is set to the state corresponding to the absence of a context switching, said flag to a state corresponding to presence of a context switching, and for enabling context switching between threads when said flag is set to correspond to the presence of context switching; and

a third means for producing a first return value when the prescribed flag is in the first

C/ state and for producing a second return value when the prescribed is in the second state, wherein said means for producing a return value is responsive and passes said return value to, said first thread making a second application program interface call requesting a termination of context switch detection; and responsive to a predetermined second application program interface call constituting a pair with said first application program interface, from said thread, for returning a value indicative of the state of said flag to said thread, said third means includes means responsive to an application program interface call from one of said plurality of threads which requests termination of detection of presence/absence of a context switching, for returning a value corresponding to the state of said flag to said thread.

a means for invalidating, for invalidating said first thread if the return value is set to said second value when said first thread calls said second application program interface.

Claim 2 (canceled)

Claim 3 (cancel):

²
Claim ~~4~~ (currently amended): The program control apparatus according to claim 1,
[[3,]] further comprising:

a means for alternately changing a priority of said first thread to high and low; and

C1 a means for comparing, for receiving a process time for said first thread between ~~from~~ said first application program interface call ~~requesting start of detection of~~ ~~presence/absence of a context switching until~~ and said second application program interface call, wherein said means for comparing ~~requesting termination of detection of~~ ~~presence/absence of a context switching, for comparing~~ compares the received process time with a remaining time until the priority of said first thread is changed to low, when said first thread is at a high priority state, and upon detection that said remaining time is shorter than said process time, lowering the priority of said first thread.

³
Claim ~~5~~ (currently amended): The program control apparatus according to claim 1, ~~[[3,]]~~ wherein said first thread is a garbage collection thread in accordance with copy method in which an object which is referenced by any other object in a memory heap area is detected, and the object is copied to a prescribed area in said heap area.

⁴
Claim ~~6~~ (currently amended): The program control apparatus according to claim 1, ~~[[3,]]~~ wherein said first thread is a memory compaction thread for eliminating fragmentation, by freeing a memory area of an object not referenced by any other object in a memory heap area as a free memory area allocatable to other object.

⁵
Claim ~~7~~ (currently amended): The ~~[[A]]~~ program control apparatus of ~~according to~~ claim 1, wherein

said first means for setting the prescribed flag includes a means responsive to the ~~an~~ first application program interface call from the first thread ~~from a thread which~~

C1
~~interface request start of detection of presence/absence of a data write to a designated memory area, for setting said a prescribed flag indicating presence/absence of a data write to said [[a]] first state, and said first state also corresponding to absence of a data write[[;]] in a designated area; and~~

said second means for setting the prescribed flag includes a means for setting, when there is a data write to said designated memory area, said prescribed flag to the [[a]] second state, and said second state also corresponding to presence of a data write. [[;]]
~~and setting said flag to another state when there is no data write to the designated memory area; and~~

~~said third means includes means responsive to an application program interface call from said thread which interface requests termination of detection of presence /absence of a data write to the designated memory area, for returning a value corresponding to the state of said flag to said thread.~~

[Claims 8-23 (canceled) ✓]

Claim ⁶24 (currently amended): A method of program control, comprising the steps of:

in response to a [[n]] first application program interface call from a first thread from a plurality of threads which requests a start of context switching detection, ~~of a presence or absence of context switching~~, setting a flag to a state indicating absence of context switching;

~~disabling context switching between threads when said flag is set to correspond to~~

~~absence of context switching;~~

C1
in response to a detection of context switching after said flag is set to the state
indicating corresponding to absence of context switching,

setting said flag to a state indicating corresponding to presence of context
switching;

~~enabling context switching between threads when said flag is set to correspond to~~
~~presence of context switching; and~~

in response to a[[n]] second application program interface call from said first thread,
~~which interface requests termination of detection of such thread. providing a return~~
value indicating the state of said flag to said first thread; and

invalidating said first thread if the return value corresponds to said flag indicating
presence of context switching.

Claim ~~25~~²⁷ (currently amended): A method of program control, comprising the steps of:

in response to a[[n]] first application program interface call from a first thread from a
plurality of threads which requests a start of detection of a presence or absence of a
data write to a designated memory area,

setting a flag to a state indicating the absence of a data write;

in response to a detection of a data write to the designated memory area after said flag
is set to the state indicating the absence of a data write,

C1
setting said flag to a state indicating corresponding to a presence of a data write;
~~when there is a data write to said designated memory area is detected; and~~
~~setting said flag to another state when there is no data write to the designated~~
~~memory area; and~~

in response to a[[n]] second application program interface call from said first thread,
~~which interfaced requests termination of detection of presence/absence of a data write~~
~~to the designed memory area;~~

returning a value corresponding to the state of said flag to said thread[[.]]; and
invalidating said first thread if the return value corresponds to said flag indicating
presence of a data write.

[Claims 26-31 (canceled)]

8
Claim 32 (currently amended): A computer readable recording medium storing a
~~program control~~ program allowing a computer to execute a program control method,
said program control method includes the steps of:

in response to a[[n]] first application program interface call from a thread which
requests a start of detection of a presence or absence of context switching,

setting a flag indicating the absence of context switching;

~~disabling context switching between threads when said flag is set to correspond~~
~~to absence of context switching;~~

C/ in response to an application program interface call from the thread, and after in response to a detection of a context switch, and after said first application program interface call,

setting said flag ~~is set to the state corresponding to the absence of context switching, setting said flag to a state corresponding to presence of context switching;~~

~~enabling context switching between threads when said flag is set to correspond to presence of context switching; and~~

in response to a[[n]] second application program interface call from said thread which requests a termination of detection of ~~the presence or absence of a context switching,~~

returning a value corresponding to the state of said flag to said thread;[[.]] and

invalidating said thread if the return value corresponds to said flag indicating presence of context switching.

9
Claim 38 (previously presented): A computer readable recording medium storing a program ~~control program~~ allowing a computer to execute a program control method, said program control method comprising the steps of:

in response to a[[n]] first application program interface call from a thread which requests a start of detection of ~~a presence or absence of a data write to a designated memory area,~~

setting a flag indicating an absence of data writes;

C1 in response to a detection of a data write to said designated memory area,

setting said flag to a state corresponding to presence of a data write; ~~when there is a data write to said designated memory area and setting said flag to another state when there is no data write to the designated memory area; and~~

in response to a[[n]] second application program interface call from said thread which ~~interface~~ requests a termination of detection of ~~presence/absence~~ of a data write to the designated memory area,

returning a value corresponding to the state of said flag to said thread;[[.]] and

D invalidating said ~~first~~ thread if the return value corresponds to said flag indicating detection of a data write.

[Claims 34-42 (canceled)